

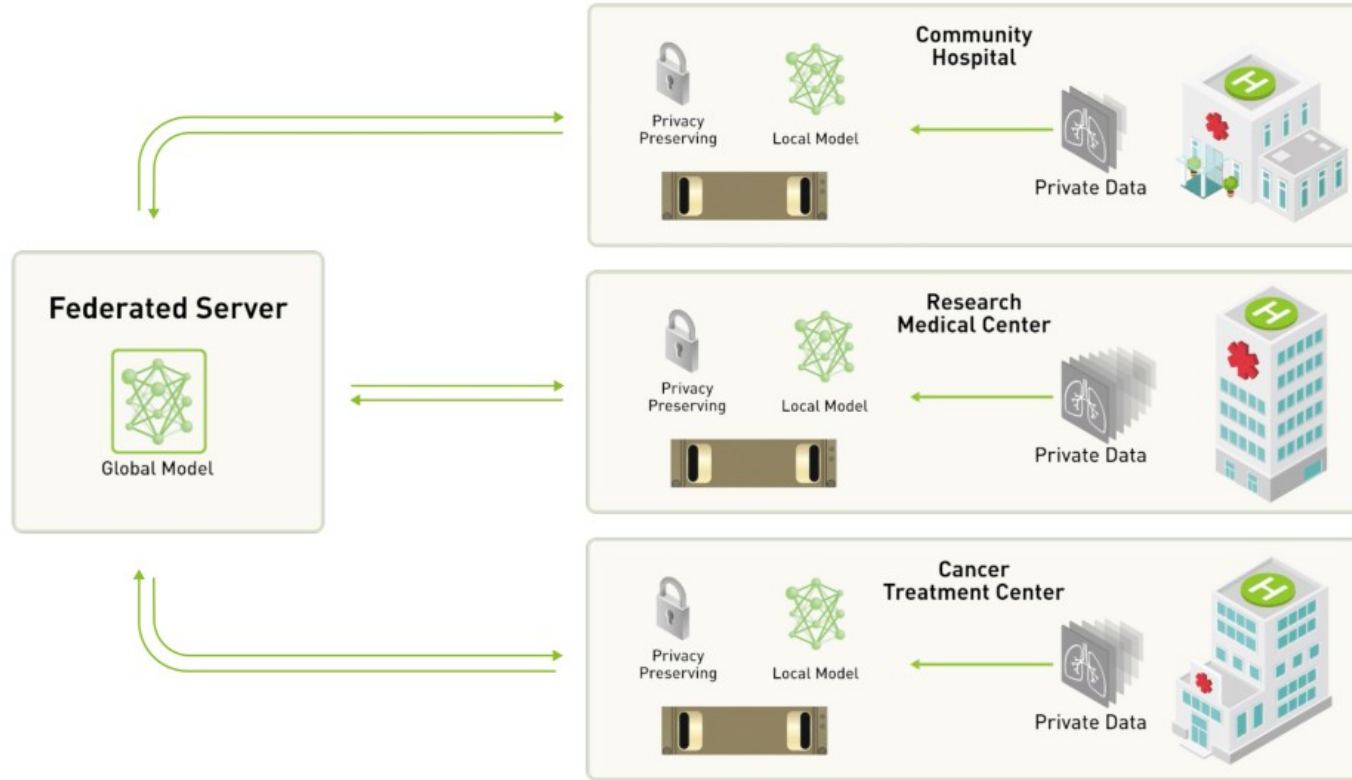


FEDERATED LEARNING

Supervisor
Rami Eisawy

Student
Muhammad Khattab

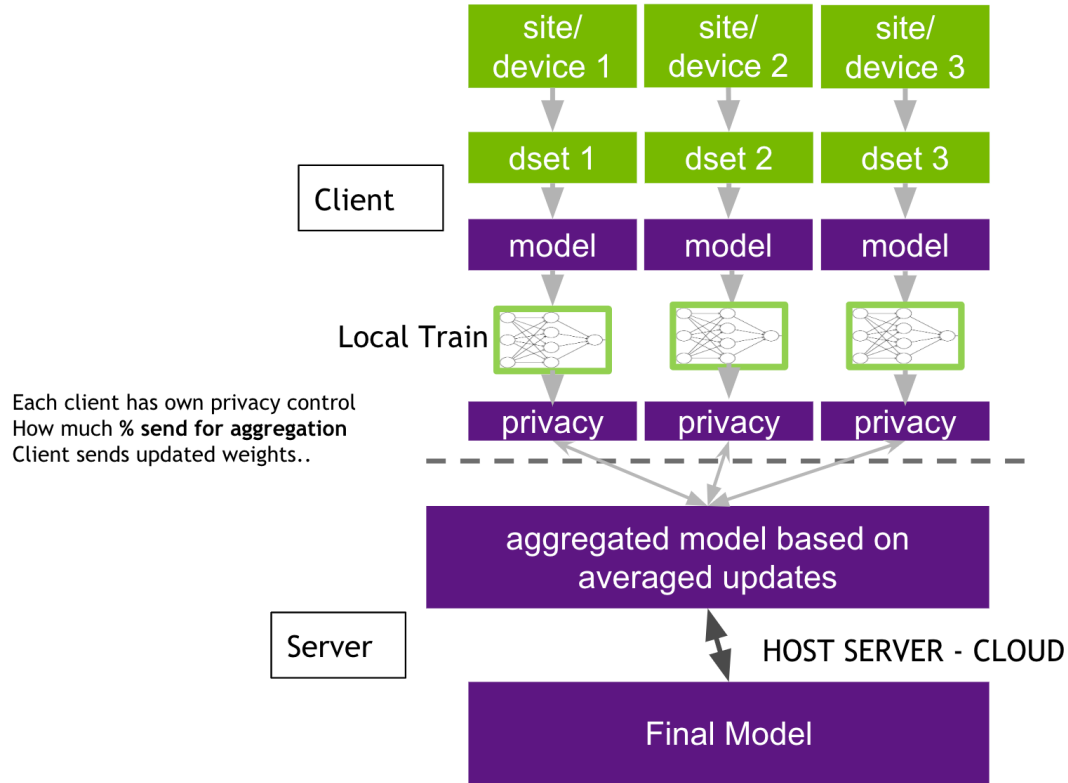
ARCHITECTURE



<https://github.com/NVIDIA/clara-train-examples/blob/master/PyTorch/NoteBooks/FL/FederatedLearning.ipynb>



CLIENT-SERVER



<https://github.com/NVIDIA/clara-train-examples/blob/master/PyTorch/NoteBooks/FL/FederatedLearning.ipynb>



CHALLENGES – TASKS (DONE)

- Switch from WSL to Ubuntu
- Install Docker – Docker Compose
- Learn about containerization
- Get a data-set
- Install Clara SDK
- Learn about framework architecture
- Get a model



CHALLENGES – TASKS (NEXT UP)

- Provisioning Federated learning Tool
- FL client Joining FL experiment
- Running FL Admin to orchestrate an FL Experiment
- Build a global model using only one client's part of data-set
- Build Global model using FedAvg aggregation scheme on all data-set
- Build global model using FedMA aggregation scheme on all data-set
- Compare Performance
- Bonus: suggest a new aggregation scheme



Ubuntu

- Windows 10 with Windows Subsystem for Linux (WSL)
- Had problems with windows updates resetting WSL
- Switched to Ubuntu 20.04



DOCKER

- managing application processes in containers
- resource-isolated processes
- similar to virtual machines
- more portable
- more resource-friendly
- more dependent on the host operating system



DOCKER - INSTALLATION

```
sudo apt update  
  
sudo apt install apt-transport-https ca-certificates curl software-properties-common  
  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  
  
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"  
  
sudo apt update  
  
apt-cache policy docker-ce  
  
sudo apt install docker-ce  
  
sudo systemctl status docker
```



DOCKER - INSTALLATION

- Option1: to use docker without `sudo`

```
sudo usermod -aG docker ${USER}
```

```
su - ${USER}
```

```
id -nG
```



DOCKER COMPOSE - INSTALLATION

- defining and running multi-container Docker applications
- use a `YAML` file to configure your application's services
- single command, you create and start all the services

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose  
  
sudo chmod +x /usr/local/bin/docker-compose
```



DATA-SET

- From Multimodal Brain Tumor Segmentation Challenge 2018 (BraTS)
- Ground truth labels, manually-revised by expert board-certified neuroradiologists
- Training, Validation, Testing:
 - Ample multi-institutional routine clinically-acquired pre-operative multimodal MRI scans of glioblastoma (GBM/HGG)
 - Lower grade glioma (LGG), with pathologically confirmed diagnosis and available OS



DATA-SET

- available as NIfTI files (.nii.gz)
- Describe:
 - native (T1)
 - post-contrast T1-weighted (T1Gd)
 - T2-weighted (T2)
 - T2 Fluid Attenuated Inversion Recovery (FLAIR)
- The overall survival (OS) data (.csv)
 - Defined in days, include the age of patients, as well as the resection status
 - only subjects with resection status of GTR (i.e., Gross Total Resection) will be evaluated
- Split the data-set over 3 clients (50%, 25%, 25%)



CLARA SDK - SETUP

```
export dockerImage=nvcr.io/nvidia/clara-train-sdk:v3.1.01  
  
docker pull $dockerImage  
  
docker run -it --rm --shm-size=1G --ulimit memlock=-1 --ulimit stack=67108864 --ipc=host --net=host  
--mount type=bind,source=/your/dataset/location,target=/workspace/data $dockerImage /bin/bash
```



MODEL - SETUP

```
ngc registry model list nvidia/med/*
```

```
MODEL_NAME=clara_mri_seg_brain_tumors_br16_full_amp  
VERSION=1
```

```
ngc registry model download-version nvidia/med/$MODEL_NAME:$VERSION --dest /var/tmp
```



MODEL - DATA

- The model is trained to segment 3 nested subregions of primary (gliomas) brain tumors
 - enhancing tumor (ET)
 - tumor core (TC)
 - whole tumor (WT)
- based on 4 input MRI scans (T1c, T1, T2, FLAIR)



MODEL – DATA

- The ET is described by areas that show hyper intensity in T1c when compared to T1, but also when compared to "healthy" white matter in T1c
- The TC describes the bulk of the tumor, which is what is typically resected
- The TC entails the ET, as well as the necrotic (fluid-filled) and the non-enhancing (solid) parts of the tumor
- The WT describes the complete extent of the disease, as it entails the TC and the peritumoral edema (ED), which is typically depicted by hyper-intense signal in FLAIR



MODEL – TRAINING CONFIGURATION

- This model utilized a similar approach described in 3D MRI brain tumor segmentation using autoencoder regularization, which was a winning method in BraTS2018
- The provided training configuration required 16GB GPU memory.
- Model Input Shape: 224 x 224 x 128
- Training Script: train.sh
- The training task uses Automatic Mixed Precision (AMP) for speed improvements.

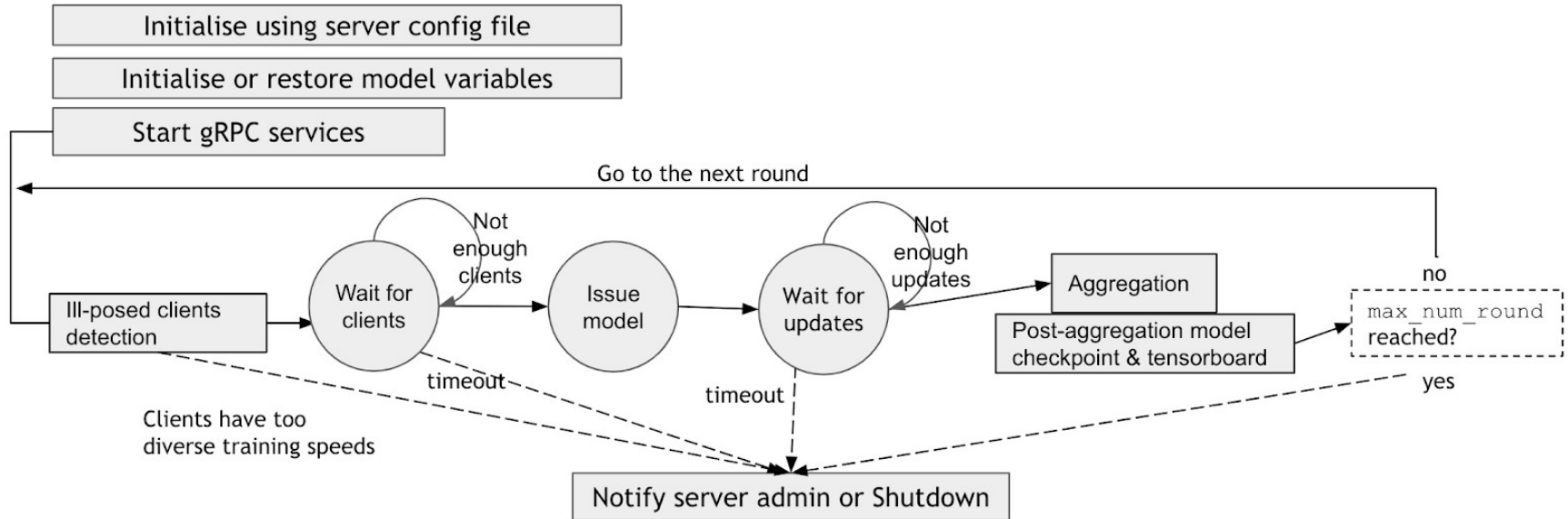


MODEL – INPUT, OUTPUT, SCORES

- Input: 4 channel 3D MRIs (T1c, T1, T2, FLAIR)
- Output: 3 channels of tumor subregion 3D masks
- Dice scores on the validation data are:
 - Tumor core (TC): 0.859
 - Whole tumor (WT): 0.904
 - Enhancing tumor (ET): 0.786

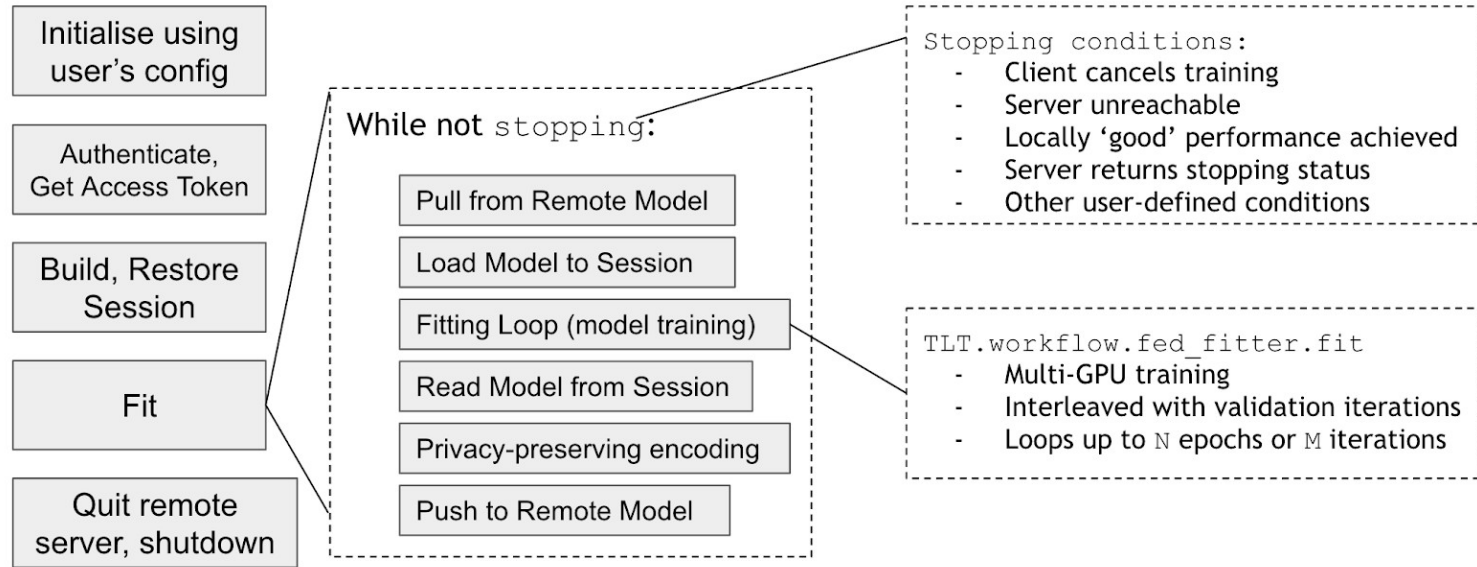


SERVER CYCLE



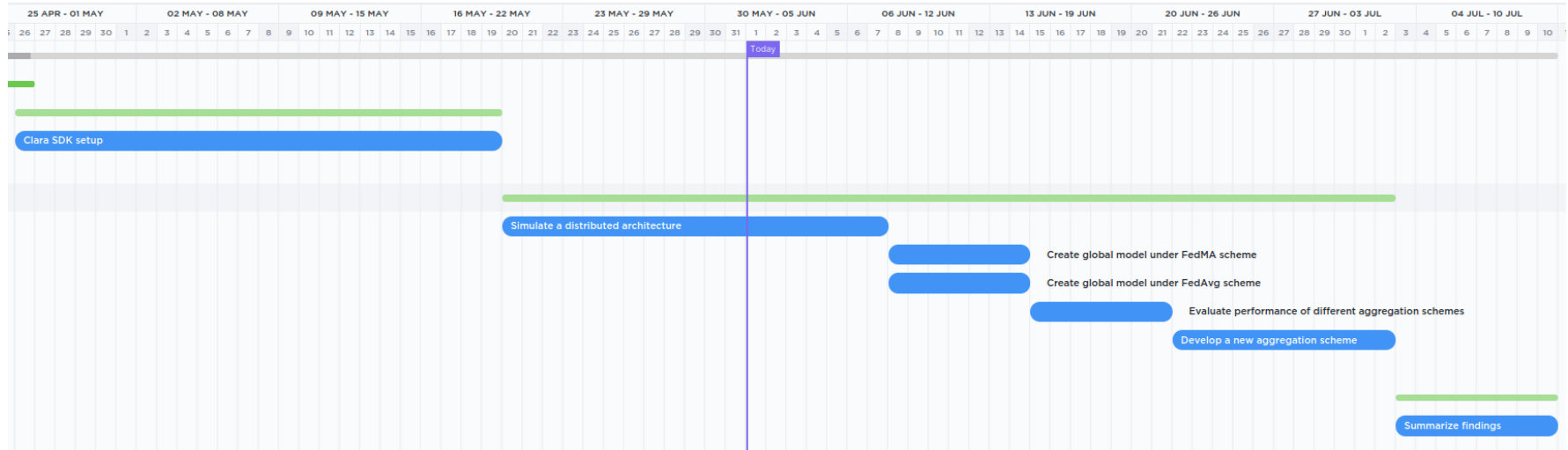
<https://github.com/NVIDIA/clara-train-examples/blob/master/PyTorch/NoteBooks/FL/FederatedLearning.ipynb>

CLIENT CYCLE



<https://github.com/NVIDIA/clara-train-examples/blob/master/PyTorch/NoteBooks/FL/FederatedLearning.ipynb>

GANTT CHART





QUESTIONS?



THANK YOU!

REFERENCES

- <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-docker-on-ubuntu-20-04>
- <https://docs.docker.com/compose/>
- https://ngc.nvidia.com/catalog/models/nvidia:med:clara_mri_seg_brain_tumors_br16_full_amp
- <https://www.med.upenn.edu/sbia/brats2018/data.html>
- <https://github.com/NVIDIA/clara-train-examples/blob/master/PyTorch/NoteBooks/FL/FederatedLearning.ipynb>

